UNIVERSITY of York

This is a repository copy of *A search-based approach to the automated design of security protocols*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/72496/

Version: Published Version

Monograph:
Hao, C, Clark, JA orcid.org/0000-0002-9230-9739 and Jacob, Jeremy Lawrence orcid.org/0000-0003-4806-7426 (2004) A search-based approach to the automated design of security protocols. Report. York Computer Science Technical Report Series ("Yellow Reports"), York Computer Scienc . Department of Computer Science, University of York

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Technical report YCS 376

A Search-based Approach to the Automated Design of Security Protocols

Chen Hao John A. Clark Jeremy L. Jacob

2004 May 5

Department of Computer Science
The University of York
YORK, YO10 5DD

Security protocols play an important role in modern communications. However, security protocol development is a delicate task; experience shows that computer security protocols are notoriously difficult to get right. Recently, Clark and Jacob provided a framework for automatic protocol generation based on combinatorial optimisation techniques and the symmetric key part of BAN logic. This paper shows how such an approach can be further developed to encompass the full BAN logic without loss of efficiency and thereby synthesise public key protocols and hybrid protocols.

Contents

List of Figures

List of Tables

1 Introduction

When we look into published security protocols, we find that many of these protocols do not succeed in their stated or implied goals. Although inappropriate use of cryptography may pose problems, it is clear that many protocols suffer problems which have nothing to do with the strength of the cryptography used. Many of the errors arise from the inappropriate structure of the message exchange. As a result, many existing protocols are susceptible to various kinds of attacks, which are independent of the weaknesses of the crypto-system employed.

Various formalisms and tools have been brought to bear on the problem. However, it would seem that automated support in this area is largely limited to the analysis and formal verification of existing protocols; there is little work in automatic protocols synthesis. The first work on automatic protocols synthesis using meta-heuristic search was presented by Clark and Jacob [CJ00, CJ01]. However, its application was limited in various ways. Most obviously principals were allowed to use only symmetric key encryption. In this paper, we show how we have extended this technique to allow public key encryption and hybrid schemes too. We also report the results of extensive experimentation with the technique.

2 Protocols and Belief Logic

Security protocols are designed to let principals communicate securely over an insecure network. Security requirements include:

Secrecy An intruder should not be able to read the contents of messages intended for others.

Authenticity If a message appears to be from Alice for an identified purpose, then Alice sent that message for that purpose.

Non-repudiation If Alice sent a message, she cannot later deny it.

Anderson and Needham show that security protocol development is a delicate task, and computer security protocols are notoriously difficult to get right [AN96]. Recent approaches to the use of formal methods in the design of security protocols include finite-state model checking and belief logics. In this paper, we concentrate on belief logics, which formalise what a principal may infer from messages received. We take as our example the first such logic, BAN [BAN89].

In 1989, Burrows, Abadi and Needham developed a belief logic (BAN logic) that could be used to reason about protocol security. Although their work has aroused much debate, the BAN logic is a milestone in the area of security protocol design and analysis. BAN logic focuses on the beliefs of honest parties involved in the protocols and on the evolution of these beliefs as a consequence of communication. The original BAN logic allows short, abstract proofs. It has identified some protocol flaws but missed others [BM93]. As a result, a number of variations and enhancements of the BAN logic have been developed. These new belief logics, such as GNY logic [GNY90] and SVO logic [SvO96], address some weaknesses of BAN logic but sacrifice its simplicity. The work described here is based on BAN, but we believe it could easily be ported to other logics. Below is a brief introduction to the notation and inference rules used in BAN logic.

2.1 Notions and Notations

2.1.1 Idealised Protocols

In much of the literature, security protocols have not been expressed in a formal manner. Such descriptions must be converted to formal descriptions if formal

analysis is to take place. In BAN logic literature the abstractions that are analysed are termed *idealised* protocols. In concrete protocols, principals maintain data items (e.g. keys) and communicate some of these items using messages with an agreed format. On receiving a message the receiver will update its state in some agreed way and carry out other agreed actions. With idealised protocols, principals maintain and communicate *beliefs*. Thus, rather than holding a key K_{ab} for session communication between A and B and distributing that key to A and B, a server S would hold the *belief* that the key K_{ab} was good for communicating between A and B (denoted $A \overset{K_{ab}}{\longleftrightarrow} B$) and include that belief in messages to A and B. The logic indicates how a receiver should update its belief state on receipt of a message.

2.1.2 Encryption and Keys

All messages in BAN logic are encrypted. Unencrypted messages sent over an insecure network provide no guarantees of any kind, because an intruder may easily alter clear-text. In practice, unencrypted concrete messages may be used as signals to cause encrypted messages to be sent, but they do not contribute to principals' beliefs. Since we work at the abstract BAN level some of our protocols have principals sending messages apparently without stimulus. Supplying such stimulus is a concrete *implementation* issue.

2.1.3 Nonce

All beliefs held in the current run of a protocol are stable for the entirety of the protocol; however, beliefs held in the past are not necessary carried forward into the present. Therefore, it is important for principals involved in a protocol to determine that messages they receive really have been created as part of the current run of the protocol. This is typically achieved by the inclusion in messages of data to bind messages to the current run. This data takes the form of numbers generated to be used only once (for bindings to the current run). These numbers used only once are commonly called *nonces*. If a principal generates a nonce for the current protocol run and receives messages that contain it, this principal may deduce that these messages have been created after the nonce was generated. An alternative to nonces are timestamps, which can also make the receiver believe the messages have been generated recently.

2.1.4 Basic Notation

The language of BAN consists of the following expressions:

Believes The assertion $P \models X$ means P *believes* the formula X. P may act as if X is true.

Sees The assertion $P \triangleleft X$ means *P sees X*. Someone has sent a message containing X to P, and P can read and repeat X; this may require decryption.

Once Said The assertion $P \mid\sim X$ means *P once said X*. The principal P at some time sent a message including the statement X. It is known that P believed X when it sent the message.

Jurisdiction The assertion $P \mid\Rightarrow X$ means *P has jurisdiction over X*. The principal P is an authority on X and should be trusted on this matter. An example of jurisdiction is that principals may believe that a key distribution server has jurisdiction over statements about the quality of keys.

Fresh The assertion $\#(X)$ means the formula X is fresh, that is to say, X has not been sent in a message at any time before the current run of the protocol. This is usually true for nonces.

Key Goodness The assertion $P \overset{K}{\longleftrightarrow} Q$ means K is a good key for communication between P and Q. That is, the key K has not been revealed to any principal other than P or Q.

Public key The assertion $\overset{K}{\longmapsto} P$ stands for the principal P having a public key K. The matching secret key (denoted by K^{-1}) will never be revealed to any principal other than P.

Secret The assertion $P \overset{X}{\rightleftharpoons} Q$ means the formula X is a secret known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. An example of a shared secret is a password.

Encryption The assertion $\{X\}_K$ means the formula X encrypted under the key K. Principals can recognise their own messages. Encrypted messages are uniquely readable and verifiable as such by holders of the right keys. Similarly, encrypted messages can be created only by a principal with the appropriate keys.

Combined The assertion $\langle X \rangle_Y$ represents X combined with the formula Y ; it is intended that Y be a secret, and that its presence prove the identity of whoever utters $\langle X \rangle_Y$.

2.2 Inference Rules

When a principal receives a message, the logic provides inference rules that indicate what new beliefs this principal may infer from the message contents. The major inference rules are given below.

Message Meaning Rules The message meaning rules explain how to derive beliefs about the origin of messages. Two of the three concern the interpretation of encrypted messages, and the third concerns the interpretation of messages with secrets.

$$\frac{P|\equiv P \overset{K}{\longleftrightarrow} Q,\ P \triangleleft \{X\}_K}{P|\equiv Q|\sim X}$$

That is, if principal P believes the key K is shared only with principal Q, and sees a message X encrypted under that key K, then P may conclude that this message X was created by Q, who 'once said' its contents X. [1]

Similarly, for public keys:

$$\frac{P|\equiv \overset{K}{\longmapsto} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P|\equiv Q|\sim X}$$

That is, if principal P believes that the key K is Q's public key and it receives a message $\{X\}_{K^{-1}}$ encrypted under Q's corresponding (private) inverse key K^{-1}, then P may conclude that principal Q once said the contents of the message.

For shared secrets:

$$\frac{P|\equiv P \overset{Y}{\rightleftharpoons} Q,\ P \triangleleft \langle X \rangle_Y}{P|\equiv Q|\sim X}$$

That is, if principal P believes that the secret Y is shared only with Q and sees $\langle X \rangle_Y$, then P believes that Q once said X.

Nonce Verification Rule The nonce verification rule expresses how a principal's view of a message changes when it determines that the message is part of the current protocol run.

$$\frac{P|\equiv \#(X),\ P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$$

That is, if P believes that X is fresh and that Q once said X, then P believes that Q has said X during the current run of protocol, and hence that Q believes X at present. In order to apply this rule, X should not contain any encrypted text. The nonce verification rule is the only way of 'promoting' once said assertion to actual belief.

Jurisdiction Rule The jurisdiction rule captures the notion that some principals are trusted to carry out certain tasks and make particular judgements.

$$\frac{P|\equiv Q|\equiv X,\ P|\equiv Q|\Rightarrow X}{P|\equiv X}$$

[1] In BAN logic, it is assumed that principals can recognise messages they themselves have created and take appropriate action when they receive such messages. We shall interpret $P \triangleleft \{X\}_K$ as "P sees message $\{X\}_K$ *and, moreover,* P knows that it did not create $\{X\}_K$ itself".

Initial Assumptions	
$S \mid\equiv \overset{K_a}{\longmapsto} A$	S believes that K_a really is A's public key
$S \mid\equiv \overset{K_b}{\longmapsto} B$	S believes that K_b really is B's public key
$S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$	S believes that K_s^{-1} is its own private key
$A \mid\equiv \overset{K_s}{\longmapsto} S$	A believes that K_s is the public key of S
$A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$	A trusts S to provide B's public key, that is, A believes that S has jurisdiction over B's public key
$A \mid\equiv N_a$	A believes that a particular number N_a is a well-formed nonce.
$A \mid\equiv \#(N_a)$	A believes that nonce N_a is actually fresh.
$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$	A believes that K_a^{-1} is its own private key.

Protocol Goal	
$A \mid\equiv \overset{K_b}{\longmapsto} B$	A believes that $\overset{K_b}{\longmapsto} B$ is B's public key

Protocol
1. $A \rightarrow S : \{N_a\}_{K_a^{-1}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}}$

Figure 2.1: Initial assumptions, a goal and a feasible protocol

That is, if principal P believes that Q believes X, and also believes that Q has jurisdiction over X, then P should believe X too.

In this paper, we also need some smaller rules, such as that $A \mid\equiv \#(X, Y)$ is deducible from $A \mid\equiv \#(X)$; we shall omit these here. Further details of these inference rules can be found in Burrows, Abadi and Needham's paper [BAN89].

2.3 Illustrative Example

Figure 2.1 gives a set of initial assumptions held by principal A and a key distribution server S, and a feasible protocol.

A believes N_a is a well formed nonce and may include it in the first message. This message is encrypted with its private key K_a^{-1}. When the server S sees (receives) this encrypted message, it can use A's public key to decrypt it and deduce $A \mid\sim N_a$, that is A once said N_a, via the *Message Meaning Rule*. Now, S may reply to A with the second message that contains two of its current beliefs: the newly derived belief A once said N_a and an initial assumption $\overset{K_b}{\longmapsto} B$. S encrypts this

message using its private key K_s^{-1}. Once A sees this message, he may decrypt it to reveal its contents. Using the *Message Meaning Rule*, A concludes $S \mid\sim \overset{K_b}{\longmapsto} B$, that is S once said K_b is B's public key. In the meantime, A may also conclude $S \mid\sim A \mid\sim N_a$, that is, S once said that A once said N_a. This message contains an assertion involving N_a, a nonce A believes to be fresh, so A may conclude the whole message is a fresh one. Then A may deduce that S believes the whole message using the *Nonce Verification Rule*. In detail, A concludes $S \mid\equiv A \mid\sim N_a$ and also $S \mid\equiv \overset{K_b}{\longmapsto} B$. Since A believes that S has jurisdiction over B's public key, A may now believe $\overset{K_b}{\longmapsto} B$ using the *Jurisdiction Rule*.

3 Search Strategy – Meta-heuristic Techniques

What we wish to do, given some assumptions and goals, is to find protocols that achieve the goals from the assumptions. That is, we wish to search the space of feasible protocols for ones satisfying a specification. Any series of honest exchanges between two or more principals defines a feasible (with respect to the logic) protocol. This is the set of feasible protocols that we consider as the design space. It is clear that this space grows exponentially as the number of messages or the number of principals rise. The choices of the belief contents of messages introduce further combinatorial complexity. For a technique to be scalable it cannot be based on simple enumeration.

Meta-heuristics are widely used to solve important practical combinatorial optimisation problems [Ree95]. The role of meta-heuristic search is to exchange guarantees of optimality for computational tractability. Examples of meta-heuristics include simulated annealing (SA)[KGV83], tabu search (TS) [AH95], genetic algorithms (GA) [Gol89], and ant colony optimisation (ACO) [BDT99]. The results reported here were obtained using simulated annealing; genetic algorithms are currently under investigation.

To use a search approach for protocol synthesis, we need to provide:

A characterisation of the design space how to represent protocols and how to distinguish valid from invalid protocols.

A fitness function if we wish to obtain the 'best' or just 'good' protocols, we must characterise precisely how 'good' a candidate protocol is.

A search strategy when exhaustive search is impractical, we need to provide a strategy for searching the design space that can locate good protocols within reasonable computational time.

We discussed the first issue in chapter 2. In this section, we show details of how to implement valid protocols within an optimisation framework and the fitness functions that we use to guide the search to a solution.

3.1 Interpreting a Message and a Move Function

A BAN protocol is represented in our search algorithm as a sequence of M messages, each of which is represented by an integer sequence. A message is sent

by one principal and received by another. N principals, indexed $0 \ldots N-1$, participate in the protocol. Associated with each of the principals is a vector of its current beliefs. Each of the M protocol messages is represented by $C+3$ integers, $vs, vr, vk, vb_1, \ldots, vb_c$. These represent the sender, the receiver, the key that the sender used to encrypt this message, and a series of C indices that reference beliefs currently held by the sending principal. So, the sender is vs mod N; receiver is vr mod N; key is vk mod $(2N + C_N^2)$ (N principals may have N private keys, N public keys, and share C_N^2 symmetric keys); the first belief in the message is belief vb_1 mod T etc., where the sender has T current beliefs, indexed $0 \ldots T-1$. $belief[0]$ is the null belief (which allows us to model easily messages with fewer than C 'real' beliefs). The vector of the receiver's current beliefs is updated after each message is sent (see below). In this way, an arbitrary sequence of integers can be interpreted as a feasible protocol (senders only ever send beliefs they actually hold). This allows a very simple move strategy for local search – simply randomly perturb any of the integers involved in any message. Although the interpreted protocol may be feasible, it may not satisfy our required goals. The fitness function, given in section 3.3 below, measures how close it comes to achieving the required goals and our search seeks to find a protocol that satisfies all these goals.

3.2 Interpreting a Protocol

This section shows how a random integer sequence can be decoded and executed as a protocol. Assume a protocol consists of M messages, each of which consists of C beliefs, and we start from the very beginning of this protocol. Firstly, we should initialise the belief state of the relevant principals involved in this protocol. Then, for each message in this protocol, we follow the steps below.

1. Determine the sender, receiver, and the key under which the current message is encrypted. If this key is an appropriate one for communication between the sender and the receiver, then proceed with the rest of the current message, else ignore this message and proceed to the next message. The method of decoding the sender, receiver, and key is indicated in section 3.1

2. Decode each of the C beliefs corresponding to current message. For instance, the first belief in the message is vb_1 mod T, where the sender currently holds T sendable beliefs.

3. Update the receiver's beliefs vector by applying the message meaning rule, the nonce verification rule, and the jurisdiction rule in that order. Here we demonstrate what a principal P will do after it receives a message $\{X, P \mid\sim N_p\}$ from another principal Q. Firstly, $Q \mid\sim X$ and $Q \mid\sim P \mid\sim N_p$ are added to P's belief vector (this represents $P \mid\equiv Q \mid\sim X$ and $P \mid\equiv Q \mid\sim P \mid\sim N_p$). This, together with (1) above implements the message meaning rule. After this, P

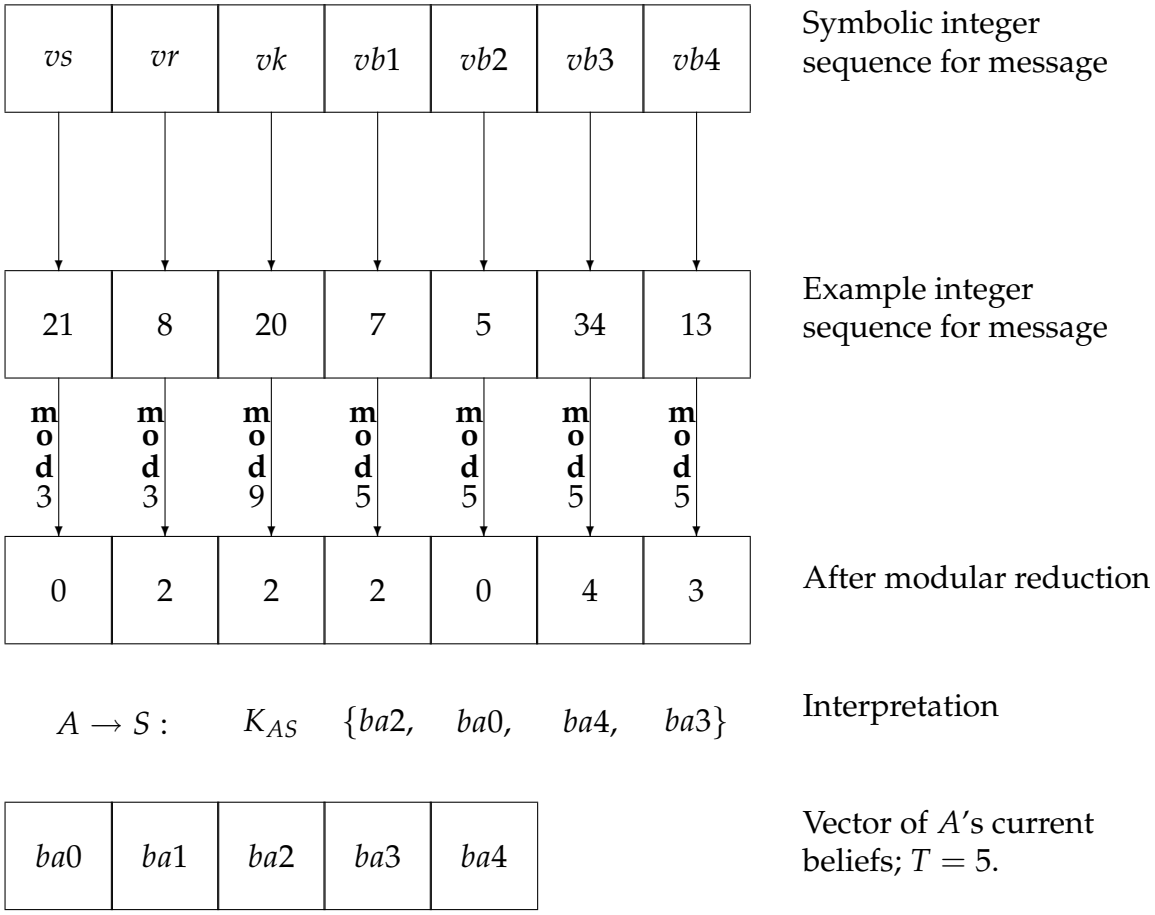

Figure 3.1: Interpreting an integer sequence. Interpretation for three principals A (= 0), B (= 1) and S (= 2). Sender A currently has 5 beliefs; '$ba0$' is the null belief. Vector of keys held omitted (there are 9 possible keys).

examines the set of received beliefs to see whether any of the beliefs contain a component that it believes to be fresh. In this case, P receives the belief $P \mid\sim N_p$, and if P believes the nonce N_p is fresh, then the whole message is regarded as fresh. If the message is fresh then the nonce verification rule is applied to add $Q \mid\equiv X$ and $Q \mid\equiv P \mid\sim N_p$. Similarly, the jurisdiction rule now may be applied to deduce further beliefs until no further beliefs can be created.

4. Record the number of required goals achieved after this message has been analysed.

Once a protocol has been executed in the above way, the fitness of this protocol can be calculated as given in the following section.

Weight	Strategy					
	EC	UC	DG	ADG	UDG	DJ
w_1	2000	500	50	0	0	0
w_2	1000	500	100	0	0	0
w_3	500	500	200	200	1000	0
w_4	200	500	500	500	1000	0
w_5	100	500	1000	1000	1000	0
w_6	50	500	2000	2000	1000	0
w_7	25	500	4000	4000	1000	1000

Table 3.1: Weighting Strategies

3.3 The Fitness Function

The fitness function is used to guide the search for a 'good' solution, that is, the fitness function must tell how 'good' a candidate solution is. We use fitness functions for a protocol of the form:

$$\sum_{i=1}^{M} w_i * g_i$$

The w_i are weightings and g_i is the number of required goals achieved after message i. In this paper, we use several weighting strategies for setting the weights w_i that are detailed in Table 3.1. These weighting strategies were first used by Clark and Jacob. Here we only introduce these weighting strategies briefly, further details of these strategies can be found in [CJ00, CJ01]. Note that the fitness function used rewards cumulatively.[1] If a goal becomes satisfied after some message it is also satisfied after all subsequent messages. Thus, the g_i form a monotone increasing sequence. The user specifies the number M of messages in a protocol.

Early Credit (EC) The weights are monotonically decreasing with i. The notion is that satisfying goals early should be rewarded.

Uniform Credit (UC) All the weights are the same.

Delayed Gratification (DG) The weights increase monotonically. This captures the idea that early satisfaction of goals may not necessarily be a good thing.

Advanced Delayed Gratification (ADG) The weights are monotonically increasing and no credit is given immediately for satisfying goals in the initial exchanges.

[1] Others approaches are clearly possible.

Uniform Delayed Gratification (UDG) No credit is given immediately for satisfying goals in the initial exchanges and later weights are equal and positive.

Destination Judgement (DJ) Only the final weights are non-zero. It does not matter how you satisfy goals, the important thing is how many you satisfy in the end.

3.4 Optimisation Techniques

In this paper we have used the well-established technique of simulated annealing [KGV83], though our implementation allows rapid interchange of optimisation techniques. The annealing approach is the standard one with a geometric cooling rate of 0.97. The number of attempted moves at each temperature is 400, with a maximum of 1000 iterations (temperature reductions) and maximum number of 50 consecutive unproductive iterations (i.e. with no move being accepted). In the interests of brevity we assume the audience is familiar with the standard annealing algorithm. A full description is given in Appendix B.

4 Experimental Method and Results

The original Clark-Jacob technique and supporting tools dealt only with symmetric key protocols. The technique and tools have now been extended to allow symmetric, public and hybrid protocols to be synthesised. This section reports the results of applying the extended technique described above to the derivation of three-party key distribution protocols. The results consist of two aspects:

1. the protocols and

2. the success fractions.

This section is organised by different sorts of protocols.

4.1 Public Key Protocols

4.1.1 Initial Assumptions

Three principals involved in this key distribution protocol are A, B and S. As a key distribution server, S holds all the other principals' public keys and its own private key. A and B both have the server's public key and their own private keys. They also maintain their own nonces that they believe to be fresh. A and B each believes that S is to be trusted on the other's public key. All the assumptions are listed below.

$S \models \xrightarrow{K_a} A, S \models \xrightarrow{K_b} B, S \models \xrightarrow{K_s^{-1}} S;$

$A \models \xrightarrow{K_s} S, A \models \xrightarrow{K_a^{-1}} A, A \models N_a, A \models \#(N_a), A \models S \models\Rightarrow \xrightarrow{K_b} B;$

$B \models \xrightarrow{K_s} S, B \models \xrightarrow{K_b^{-1}} B, B \models N_b, B \models \#(N_b), B \models S \models\Rightarrow \xrightarrow{K_a} A.$

4.1.2 Goals

At the end of the protocol run, both A and B must believe that they hold each other's public key. The other two goals require that each of them believes the other believes its public key is good.

$A \models \xrightarrow{K_b} B, B \models \xrightarrow{K_a} A;$

$A \models B \models \xrightarrow{K_a} A, B \models A \models \xrightarrow{K_b} B.$

1. $A \rightarrow S : \{N_a\}_{K_a^{-1}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}}$
3. $B \rightarrow S : \{N_b\}_{K_b^{-1}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}}$
5. $B \rightarrow A : \{N_b\}_{K_b^{-1}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, \overset{K_b}{\longmapsto} B \right\}_{K_a^{-1}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, \overset{K_a}{\longmapsto} A \right\}_{K_b^{-1}}$

Figure 4.1: Public key protocol generated during experimentation

4.1.3 Results and Statistics

Twenty runs of the program were carried out for each fitness function strategy. In our program, the annealing parameters given in section 3.4 were used. Figure 4.1 shows one of the public key protocols generated by the program.

In the rest of this paper, only the core security relevant components of a protocol are presented. That is, our descriptions of protocols do not include those belief components that do not contribute to the predefined goals. In addition, redundant beliefs (where the same beliefs are included twice or more in one message) have also been removed. Currently, these 'junk' beliefs are removed by hand; automating their removal is under investigation.

The search rapidly established the first two beliefs $A \mid\equiv \overset{K_b}{\longmapsto} B$ and $B \mid\equiv \overset{K_a}{\longmapsto} A$ after 4 messages. In order to achieve the third goal $B \mid\equiv A \mid\equiv \overset{K_b}{\longmapsto} B$, A must have knowledge of the nonce N_b and show it to B (A sends B a message including $B \mid\sim N_b$). In our program, sendable beliefs are either simple (e.g. N_a, $\overset{K_a}{\longmapsto} A$ are both sendable) or else involve only one operator (e.g. $A \mid\sim N_a$, $S \mid\Rightarrow \overset{K_a}{\longmapsto} A$ and so on). The only way A can acquire knowledge of N_b is to receive it in a message from B directly. (It would be possible to obtain knowledge of N_b via S, but this would require additional assumptions, such as $A \mid\equiv S \mid\Rightarrow B \mid\sim N_b$.)

When we input $A \mid\equiv S \mid\Rightarrow B \mid\sim N_b$ and $B \mid\equiv S \mid\Rightarrow A \mid\sim N_a$ as two initial assumptions, our program generates protocols that achieve our goals (same as before) within 6 messages. Figure 4.2 shows one of these protocols.

From an abstract logic point of view, the more beliefs included in messages the

1. $A \rightarrow S : \{N_a\}_{K_a^{-1}}$
2. $B \rightarrow S : \{N_b\}_{K_b^{-1}}$
3. $S \rightarrow A : \left\{ A \mid\sim N_a, B \mid\sim N_b, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, A \mid\sim N_a, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}}$
5. $B \rightarrow A : \left\{ A \mid\sim N_a, \overset{K_a}{\longmapsto} A \right\}_{K_b^{-1}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, \overset{K_b}{\longmapsto} B \right\}_{K_a^{-1}}$

Figure 4.2: Public key protocol with additional assumptions

Strategy	Success Fraction Four Beliefs Per Message	Success Fraction Three Beliefs Per Message
EC	0.70	0.40
UC	0.80	0.60
DG	0.90	0.60
ADG	0.85	0.60
UDG	0.80	0.40
DJ	0.05	0.05

Table 4.1: Effect on success fraction of varying numbers of fields per message

more information the receiving principal obtains. Thus, messages with more information create a greater probability of achieving goals. We have also repeated the above experiments allowing three beliefs per message. Table 4.1 gives the success fractions when four and three beliefs per message are used. As we have seen, for the original problem, in all cases except for the destination judgement (DJ), the success fractions are decreased dramatically. Another conclusion we can draw from these figures is that appropriate redundancy is actually very useful to the optimisation approach. Moreover, DJ is clearly awful in both cases. Some degree of reward for early achievement is clearly useful.

4.2 Public Key Protocols using Timestamps

4.2.1 Initial Assumptions

Again, three principals involved in this sort of key distribution protocol are A, B and S, where S is a trustworthy key distribution server. Here we allow the notion of timestamps. T is, effectively, a form of nonce shared by all parties prior to the run of the protocol. The difference is this sort of protocol relies heavily on synchronised clocks, since each principal believes that a timestamp generated elsewhere is fresh (if it has a value within a window of the receiver's local time).

$A \mid\equiv \stackrel{K_s}{\longmapsto} S$, $A \mid\equiv \stackrel{K_a^{-1}}{\longmapsto} A$, $A \mid\equiv T$, $A \mid\equiv \#(T)$,
$A \mid\equiv S \mid\Rightarrow \stackrel{K_b}{\longmapsto} B$;
$B \mid\equiv \stackrel{K_s}{\longmapsto} S$, $B \mid\equiv \stackrel{K_b^{-1}}{\longmapsto} B$, $B \mid\equiv T$, $B \mid\equiv \#(T)$,
$B \mid\equiv S \mid\Rightarrow \stackrel{K_a}{\longmapsto} A$;
$S \mid\equiv \stackrel{K_a}{\longmapsto} A$, $S \mid\equiv \stackrel{K_b}{\longmapsto} B$, $S \mid\equiv \stackrel{K_s^{-1}}{\longmapsto} S$, $S \mid\equiv T$,
$S \mid\equiv \#(T)$.

4.2.2 Goals

We hope this sort of protocol can achieve the following four goals (as before).

$A \mid\equiv \stackrel{K_b}{\longmapsto} B$, $B \mid\equiv \stackrel{K_a}{\longmapsto} A$;
$A \mid\equiv B \mid\equiv \stackrel{K_a}{\longmapsto} A$, $B \mid\equiv A \mid\equiv \stackrel{K_b}{\longmapsto} B$.

4.2.3 Results and Statistics

We use the same annealing parameters as those used in subsection 4.1.3. Figure 4.3 shows one of the protocols generated by our program, and Table 4.2 shows the success fraction for each search strategy when we allow four beliefs in each message.

1. $S \rightarrow A : \left\{ T, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}}$
2. $S \rightarrow B : \left\{ T, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}}$
3. $B \rightarrow A : \left\{ T, \overset{K_a}{\longmapsto} A \right\}_{K_b^{-1}}$
4. $A \rightarrow B : \left\{ T, \overset{K_b}{\longmapsto} B \right\}_{K_a^{-1}}$

Figure 4.3: Public key protocol using timestamps, four beliefs per message

Strategy	Success Fraction
EC	0.95
UC	1.00
DG	0.90
ADG	0.65
UDG	0.85
DJ	0.60

Table 4.2: Success fractions for protocols using timestamps

4.3 Hybrid Protocols

We now attempt to evolve a protocol allowing the use of both symmetric and public key encryption.

4.3.1 Initial Assumptions

Essentially, we aim to distribute a secret key using public key means. In this sort of protocol, we assume that both principals A and B can communicate with the server S via a public key. The server S will distribute a symmetric session key that will be used in further communications between A and B.

$A \mid\equiv \stackrel{K_s}{\longmapsto} S$, $A \mid\equiv \stackrel{K_a^{-1}}{\longmapsto} A$, $A \mid\equiv N_a$, $A \mid\equiv \#(N_a)$,

$A \mid\equiv S \mid\Rightarrow A \stackrel{K_{ab}}{\longleftrightarrow} B$;

$B \mid\equiv \stackrel{K_s}{\longmapsto} S$, $B \mid\equiv \stackrel{K_b^{-1}}{\longmapsto} B$, $B \mid\equiv N_b$, $B \mid\equiv \#(N_b)$,

$B \mid\equiv S \mid\Rightarrow A \stackrel{K_{ab}}{\longleftrightarrow} B$;

$S \mid\equiv \stackrel{K_a}{\longmapsto} A$, $S \mid\equiv \stackrel{K_b}{\longmapsto} B$, $S \mid\equiv \stackrel{K_s^{-1}}{\longmapsto} S$,

$S \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$.

4.3.2 Goals

The four desired goals are:

$A \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$, $B \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$;

$A \mid\equiv B \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$, $B \mid\equiv A \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$.

4.3.3 Results and Statistics

Figure 4.4 shows one of the hybrid protocols generated by the program. Table 4.3 shows the success fraction for each search strategy when we allow four beliefs in one message. When considering protocols that distribute secret information, we assume that any signed messages are then encrypted with the public key of the receiver as shown in Figure 4.4. This is a sound assumption that is recommended by Anderson and Needham [AN96]. However, we can see that the nonce need not be kept secret and so the outer encryptions (using K_s) in message 1 and 3 may be removed. We can see that (default) other encryptions in message 2 and 4 are necessary (to maintain confidentiality of the shared key K_{ab}).

4.4 Extensive Experimentation

We have applied our approach to the on-line repository of security protocols "Security Protocols Open Repository" at http://www.lsv.ens-cachan.fr/spore/, which

1. $A \rightarrow S : \left\{ \{N_a\}_{K_a^{-1}} \right\}_{K_s}$
2. $S \rightarrow A : \left\{ \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_s^{-1}} \right\}_{K_a}$
3. $B \rightarrow S : \left\{ \{N_b\}_{K_b^{-1}} \right\}_{K_s}$
4. $S \rightarrow B : \left\{ \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_s^{-1}} \right\}_{K_b}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

Figure 4.4: A hybrid encryption protocol

Strategy	Success Fraction
EC	0.80
UC	0.85
DG	0.85
ADG	0.60
UDG	0.80
DJ	0.05

Table 4.3: Success fractions for hybrid protocols

contains 45 protocols. Our program successfully synthesised 38 BAN protocols when given the same initial assumptions and goals as they are in the original ones. Obviously, all these 38 protocols are correct according to the BAN logic, and the successful synthesis process itself is a proof. The remaining 7 sets of assumptions and goals contain features outside of the BAN logic, and so our tool cannot be used. There are several "variations on a theme" in the library. Often, the assumptions and goals of these variants are the same. Thus, there may be one abstract specification and several concrete implementations. For presentational consistency, we have presented the concrete variants and have simply repeated the abstract specification and the corresponding protocol we have evolved. This gives the reader some feel for how abstract requirements can be satisfied in many different concrete ways (and allows the reader to read each protocol individually). The 38 concrete protocols presented in the appendix correspond to 23 distinct abstract specifications.

Experimentation highlighted difficulties with repeated authentication. In a typical repeated authentication protocol, the first (preliminary) part typically distributes a 'ticket' to a principal. This ticket can be used by the principal to achieve some authentication task. This may be described as the full initial run of the protocol. However, the ticket will typically be used several more times (within a specified lifetime). A well-known repeated authentication protocol is the Neumann Stubblebine protocol given in section A.17.

The logic (and so our tools) has difficulties with the use of repeated 'tickets'. We have simply evolved protocols to meet the goals of the first authentication run. It is generally possible to address the repeated parts of the protocol, provided each presentation of a ticket is regarded as 'fresh enough' (or simply 'fresh') if its lifetime has not expired. With such an approach, the evolution of mutual authenticating nonce exchanges is generally trivial. Nevertheless, our tools currently do not allow keys of the form K_{pp} (a key known only to P and used to seal tickets to be presented repeatedly to P).

5 Conclusions and Further Work

The above work shows that the original Clark-Jacob approach for the symmetric case can be successfully extended to allow public key and hybrid cryptographic schemes. The protocols generated, although simple, are typical abstractions of protocols in the literature. The ease with which the approach generated protocols satisfying realistic goals merits further investigation of the technique. Experimentation and our general knowledge of protocol verification techniques have allowed us to identify numerous possible improvements to the approach and tool support. These are outlined below.

A more sophisticated logic (SVO seems a promising candidate) should be adopted to increase design choice and give greater confidence in the practical security of evolved protocols. As far as actual freedom from security flaws is concerned, we are very much at the mercy of the logic we choose. We have expanded the previously used subset of BAN logic to allow public key and hybrid protocols to be evolved. We need now to address weaknesses in the BAN logic itself. Similarly, allowing more sophisticated beliefs to be communicated in messages should allow a wider range of protocols to be evolved.

Non-functional properties such as efficiency are an important consideration for most security protocol designers and should be incorporated into our design synthesis approach. Efficiency has not been ignored completely in the current approach — the cumulative reward nature of the fitness function generally favours shorter protocols — but this is somewhat indirect and does not address crucial issues such as amount of encryption etc. Automatic refinement to a more detailed representation (code, for example) would be a significant enhancement and would greatly facilitate inclusion of non-functional issues.

We may need to exploit the approach taken to the fullest extent possible. The model checking approaches [PS00a, PS00b] are distinctly limited, e.g. three or four messages, in the size of the protocols they can produce. In this paper we have presented seven-message protocols and some nine-message protocols were demonstrated by Clark and Jacob[CJ01]. We currently do not know the limits of the optimisation approaches. Earlier work [CJ01] showed that the protocol generation problem suffers from combinatorial explosion. As the complexity of the underlying logic increases, so does the magnitude of the search problem. Simulated annealing and genetic algorithms may not be the best optimisation techniques to use. Others should be considered.

The work here shows that meta-heuristic search approaches to secure protocol synthesis are potentially powerful and have the benefit of being rapid. Our tools could generate candidate protocols rapidly and concrete refinements of them

could be subjected to more detailed and sophisticated analysis (such as that provided by current model checking approaches). This would provide an interesting synthesis of current techniques.

Our experiments have tested the approach's ability to generate protocols for existing and fairly standard requirements. It seems suited to such tasks. It will be interesting to see whether meta-heuristic approaches will be able to produce protocols for novel or highly complex requirements. Will there be any surprises?

Bibliography

[AH95] D. de Werra A. Hertz, E. Taillard. A tutorial on tabu search. In *Proc. of Giornate di Lavoro AIRO'95 (Enterprise Systems: Management of Technological and Organizational Changes)*, pages 13–24, Italy, 1995.

[AK88] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, 1988.

[AN95] Ross Anderson and Roger Needham. Robustness principles for public key protocols. *Advances in Cryptology - CRYPTO '95: 15th Annual International Cryptology Conference, Lecture Notes in Computer Science*, 963:236–247, 1995.

[AN96] Martín Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, 1996.

[And01] Ross Anderson. *Security Engineering*. John Wiley & Sons, 2001.

[BAN89] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. Technical Report 39, Digital Systems Research Center, February 1989.

[BAN90] Michael Burrows, Martín Abadi, and Roger Needham. The scope of a logic of authentication. Technical Report 39, Digital Systems Research Center, February 1990.

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies on the Sciences of Complexity)*. Oxford University Press Inc, USA; ISBN: 0195131592, 1999. ISBN: 0195131592.

[BM93] Colin Boyd and Wenbo Mao. On a limitation of BAN logic. In *Advances in Cryptology - EUROCRYPT '93: Workshop on the Theory and Application of Cryptographic Techniques, Lecture Notes in Computer Science*, volume 765, pages 240–247, 1993.

[CJ97] John A. Clark and Jeremy L. Jacob. A survey of authentication protocol literature, 1997. Available as http://www-users.cs.york.ac.uk/~jac/papers/drareview.ps.gz.

[CJ00] John A. Clark and Jeremy L. Jacob. Search for a solution: Engineering tradeoffs and the evolution of provably secure protocols. In *Proceedings of 2000 IEEE Symposium on Research in Security and Privacy*, pages 82–95. IEEE Computer Society, May 2000.

[CJ01] John A. Clark and Jeremy L. Jacob. Protocols are programs too: the meta-heuristic search for security protocols. *Information and Software Technology*, 43(14):891–904, December 2001.

[Cla02] John A. Clark. *Metaheuristic Search as a Cryptological Tool*. PhD thesis, University of York, 2002.

[DS81] Dorothy Denning and G. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8), August 1981.

[GNY90] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of 1990 IEEE Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society, May 1990.

[Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[Gon91] Li Gong. Handling infeasible specifications of cryptographic protocols. In *Proceedings of The 4th IEEE Computer Security Foundations Workshop*, pages 99–102. IEEE Computer Society, June 1991.

[HLS00] James Heather, Gavin Lowe, and Steve Schneider. How to prevent type flaw attacks on security protocols. In *Proceedings of The 13th Computer Security Foundations Workshop*, pages 32–43. IEEE Computer Society Press, July 2000.

[KGV83] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[Low99] Gavin Lowe. Towards a completeness result for model checking of security protocols. In *Proceedings of The 11th Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1999.

[MCJ97] Will Marrero, Edmund Clarke, and Somesh Jha. Model checking for security protocols. Technical Report CMU-CS-97-139, Carnegie Mellon University, May 1997.

[MSN94] A. Mathuria, R. SafaviNaini, and P. Nickolas. Some remarks on the logic of Gong, Needham and Yahalom. In *Proceedings of International Computer Symposium 1994*, volume 1, pages 303–308, December 1994.

[MSN95] A. Mathuria, R. SafaviNaini, and P. Nickolas. On the automation of GNY logic. In *Proceedings of the 18th Australian Computer Science Conference*, volume 17, pages 370–379, February 1995.

[NS78] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), December 1978.

[PS00a] Adrian Perrig and Dawn Song. A first step towards the automatic generation of security protocols. In *Proceedings of Network and Distributed System Security 2000*, pages 73–83, February 2000.

[PS00b] Adrian Perrig and Dawn Song. Looking for diamonds in the desert – extending automatic protocol generation to three-party authentication and key agreement protocols. In *Proceedings of The 13th IEEE Computer Security Foundations Workshop*. IEEE Computer Society, July 2000.

[Ree95] Colin Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. McGraw-Hill Book Company Europe, 1995.

[SC01] Paul Syverson and Iliano Cervesato. The logic of authentication protocols. *Foundations of Security Analysis and Design : Tutorial Lectures, Lecture Notes in Computer Science*, 2171:63–136, 2001.

[Sch96] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, second edition, 1996.

[SvO96] Paul F. Syverson and Paul C. van Oorschot. A unified cryptographic protocol logic, 1996. NRL Publication 5540–227, Naval Research Lab.

[Vas01] Boris W. Vassall. A look at automatic protocol generation & security protocols, July 2001. Available as http://www.sans.org/rr/protocols/auto.php.

A Results on Protocols Library

This appendix documents the results of our successful attempts to synthesise protocols from the Clark-Jacob protocol library [CJ97]. For each of the 38 protocols considered we first present the protocol in 'standard notation', reflecting the data contents of messages in successful completing protocols. We then provide the assumptions and goals (i.e. the requirements specification) of the protocol expressed in BAN logic. These have been reverse engineered by the authors. They seem highly plausible given our knowledge of protocols in general and of examples in the literature where BAN logic has been applied. We then present the belief logic representation of one of the protocols evolved to meet the specification using our automated heuristic search procedures. Occasionally, additional comments are provided for clarification of important issues.

A.1 Andrew Secure RPC

1. $A \rightarrow B : A, \{N_a\}_{K_{ab}}$
2. $B \rightarrow A : \{N_a + 1, N_b\}_{K_{ab}}$
3. $A \rightarrow B : \{N_b + 1\}_{K_{ab}}$
4. $B \rightarrow A : \{K'_{ab}\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv B \mid\Rightarrow A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv A \mid\equiv B \mid\sim N_b$

Comment The protocol fails to satisfy its desired goals. After a run of this protocol, each principal is aware that the other exists, but no more. (We use $B \models\equiv A \models\equiv B \models\sim N_b$ to ensure that B knows A exists. Actually, B ought to be ensured that A really exists before B generates and sends out a new session key. However, BAN and Lowe ignored this condition when they modified this protocol.)

BAN Protocol generated by our program

1. $B \rightarrow A : \{N_b\}_{K_{ab}}$
2. $A \rightarrow B : \{N_a, B \mid\sim N_b\}_{K_{ab}}$
3. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K'_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.2 BAN Modified Andrew Secure RPC

1. $A \rightarrow B : A, \{N_a\}_{K_{ab}}$
2. $B \rightarrow A : \{N_a + 1, N_b\}_{K_{ab}}$
3. $A \rightarrow B : \{N_b + 1\}_{K_{ab}}$
4. $B \rightarrow A : \{K'_{ab}, N_a\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv B \mid\Rightarrow A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv A \mid\equiv B \mid\sim N_b$

Comment Compared to the original protocol, this modified version does not modify the initial assumptions and desired goals. For this reason, our program gives the same solution.

BAN Protocol generated by our program

1. $B \rightarrow A : \{N_b\}_{K_{ab}}$
2. $A \rightarrow B : \{N_a, B \mid\sim N_b\}_{K_{ab}}$
3. $B \rightarrow A : \left\{A \mid\sim N_a, A \overset{K'_{ab}}{\longleftrightarrow} B\right\}_{K_{ab}}$

A.3 BAN Concrete Andrew Secure RPC

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow A : \{N_a, K'_{ab}\}_{K_{ab}}$
3. $A \rightarrow B : \{N_a\}_{K'_{ab}}$
4. $B \rightarrow A : N_b$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv B \mid\Rightarrow A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K'_{ab}}{\longleftrightarrow} B)$
Desired Goals	Standard	$A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow B : \{N_a\}_{K_{ab}}$
2. $B \rightarrow A : \left\{A \mid\sim N_a, A \overset{K'_{ab}}{\longleftrightarrow} B\right\}_{K_{ab}}$
3. $A \rightarrow B : \left\{A \overset{K'_{ab}}{\longleftrightarrow} B\right\}_{K_{ab}}$

A.4 Lowe Modified Andrew Secure RPC

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow A : \{N_a, K'_{ab}, B\}_{K_{ab}}$
3. $A \rightarrow B : \{N_a\}_{K'_{ab}}$
4. $B \rightarrow A : N_b$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv B \mid\Rightarrow A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K'_{ab}}{\longleftrightarrow} B)$
Desired Goals	Standard	$A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program (same as before)

1. $A \rightarrow B : \{N_a\}_{K_{ab}}$
2. $B \rightarrow A : \left\{A \mid\sim N_a, A \overset{K'_{ab}}{\longleftrightarrow} B\right\}_{K_{ab}}$
3. $A \rightarrow B : \left\{A \overset{K'_{ab}}{\longleftrightarrow} B\right\}_{K_{ab}}$

A.5 CCITT X.509 (1)

1. $A \rightarrow B : A, \left\{ T_a, N_a, B, X_a, \{Y_a\}_{K_b} \right\}_{K_a^{-1}}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv T_a$ $A \mid\equiv \#(T_a)$ $B \mid\equiv T_a$ $B \mid\equiv \#(T_a)$ $A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $A \mid\equiv X_a$
	Additional	$A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv X_a$

Comment T_a used here means a timestamp generated by principal A. When B receives it, B will check whether this timestamp is 'fresh' or not. We use $B \mid\equiv \#(T_a)$ to indicate this point.

We can see that the two additional assumptions, which are used in the original protocol, seem no obvious security purpose.

BAN Protocol generated by our program

1. $A \rightarrow B : \left\{ \left\{ T_a, X_a, A \overset{Y_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$

A.6 CCITT X.509 (1c)

1. $A \rightarrow B : A, \left\{ T_a, N_a, B, X_a, \left\{ Y_a, \{h(Y_a)\}_{K_a^{-1}} \right\}_{K_b} \right\}_{K_a^{-1}}$

Assumptions	Standard	$A \mid\equiv \stackrel{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \stackrel{K_b}{\longmapsto} B$ $B \mid\equiv \stackrel{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \stackrel{K_a}{\longmapsto} A$ $A \mid\equiv T_a$ $A \mid\equiv \#(T_a)$ $B \mid\equiv T_a$ $B \mid\equiv \#(T_a)$ $A \mid\equiv A \stackrel{Y_a}{\rightleftharpoons} B$ $A \mid\equiv X_a$
	Additional	$A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv A \stackrel{Y_a}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv X_a$

BAN Protocol generated by our program (same as before)

1. $A \rightarrow B : \left\{ \left\{ T_a, X_a, A \stackrel{Y_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$

A.7 CCITT X.509 (3)

1. $A \rightarrow B : A, \left\{ T_a, N_a, B, X_a, \{Y_a\}_{K_b} \right\}_{K_a^{-1}}$
2. $B \rightarrow A : B, \left\{ T_b, N_b, A, N_a, X_b, \{Y_b\}_{K_a} \right\}_{K_b^{-1}}$
3. $A \rightarrow B : A, \{N_b\}_{K_a^{-1}}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv T_a$ $A \mid\equiv \#(T_a)$ $A \mid\equiv T_b$ $A \mid\equiv \#(T_b)$ $B \mid\equiv T_a$ $B \mid\equiv \#(T_a)$ $B \mid\equiv T_b$ $B \mid\equiv \#(T_b)$ $A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $B \mid\equiv A \overset{Y_b}{\rightleftharpoons} B$ $A \mid\equiv X_a$ $B \mid\equiv X_b$
	Additional	$A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
Desired Goals	Standard	$A \mid\equiv B \mid\equiv A \overset{Y_b}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $A \mid\equiv B \mid\equiv X_b$ $B \mid\equiv A \mid\equiv X_a$

BAN Protocol generated by our program

1. $A \rightarrow B : \left\{ \left\{ T_a, X_a, A \overset{Y_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$
2. $B \rightarrow A : \left\{ \left\{ T_b, X_b, A \overset{Y_b}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$

Comment The outer encryption for Message 1 is unnecessary, and maybe removed (N_a is not a secret).

A.8 BAN modified version of CCITT X.509 (3)

1. $A \rightarrow B : A, \left\{ N_a, B, X_a, \{Y_a\}_{K_b} \right\}_{K_a^{-1}}$
2. $B \rightarrow A : B, \left\{ N_b, A, N_a, X_b, \{Y_b\}_{K_a} \right\}_{K_b^{-1}}$
3. $A \rightarrow B : A, \{B, N_b\}_{K_a^{-1}}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $B \mid\equiv A \overset{Y_b}{\rightleftharpoons} B$ $A \mid\equiv X_a$ $B \mid\equiv X_b$
	Additional	$A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
Desired Goals	Standard	$A \mid\equiv B \mid\equiv A \overset{Y_b}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{Y_a}{\rightleftharpoons} B$ $A \mid\equiv B \mid\equiv X_b$ $B \mid\equiv A \mid\equiv X_a$

BAN Protocol generated by our program

1. $A \rightarrow B : \left\{ \{N_a\}_{K_a^{-1}} \right\}_{K_b}$
2. $B \rightarrow A : \left\{ \left\{ A \mid\sim N_a, N_b, X_b, A \overset{Y_b}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$
3. $A \rightarrow B : \left\{ \left\{ B \mid\sim N_b, X_a, A \overset{Y_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$

A.9 Denning-Sacco Shared Key

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \left\{ B, K_{ab}, T, \{K_{ab}, A, T\}_{K_{bs}} \right\}_{K_{as}}$
3. $A \rightarrow B : \{K_{ab}, A, T\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv T$ $A \mid\equiv \#(T)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv T$ $B \mid\equiv \#(T)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv T$ $S \mid\equiv \#(T)$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
2. $S \rightarrow B : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$

A.10 Lowe Modified Denning-Sacco Shared Key

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \left\{ B, K_{ab}, T, \{K_{ab}, A, T\}_{K_{bs}} \right\}_{K_{as}}$
3. $A \rightarrow B : \{K_{ab}, A, T\}_{K_{bs}}$
4. $B \rightarrow A : \{N_b\}_{K_{ab}}$
5. $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv T$ $A \mid\equiv \#(T)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv T$ $B \mid\equiv \#(T)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv T$ $S \mid\equiv \#(T)$
	Additional	$B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment The additional goal is certainly not met by the protocol because A cannot recognise the nonce N_b. It is interesting to note that our evolved protocol uses only timestamps (ignores the nonce N_b).

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
2. $S \rightarrow B : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
3. $B \rightarrow A : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}} from\ B$
4. $A \rightarrow B : \left\{ T, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}} from\ A$

A.11 Kao Chow Authentication v.1

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow B : \{A, B, N_a, K_{ab}\}_{K_{as}}, \{A, B, N_a, K_{ab}\}_{K_{bs}}$
3. $B \rightarrow A : \{A, B, N_a, K_{ab}\}_{K_{as}}, \{N_a\}_{K_{ab}}, N_b$
4. $A \rightarrow B : \{N_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_a$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment Obviously, the additional assumption $B \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ might cause a replay/freshness attack.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $B \rightarrow A : \left\{ N_b, A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
5. $A \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.12 Kao Chow Authentication v.2

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow B : \{A, B, N_a, K_{ab}, K_t\}_{K_{as}}, \{A, B, N_a, K_{ab}, K_t\}_{K_{bs}}$
3. $B \rightarrow A : B, \{A, B, N_a, K_{ab}, K_t\}_{K_{as}}, \{N_a, K_{ab}\}_{K_t}, N_b$
4. $A \rightarrow B : \{N_b, K_{ab}\}_{K_t}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_t}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K_t}{\longleftrightarrow} B)$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_a$ $A \mid\equiv S \mid\Rightarrow A \overset{K_t}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \overset{K_t}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv A \overset{K_t}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_t}{\longleftrightarrow} B$

Comment K_t is an additional fresh symmetric key whose purpose is to prevent a freshness attack in Kao Chow Authentication v.1. However, we can see that the additional assumption $B \mid\equiv \#(A \overset{K_t}{\longleftrightarrow} B)$ might cause a replay/freshness attack as well as the first version of this protocol.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B, A \overset{K_t}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B, A \overset{K_t}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $B \rightarrow A : \left\{ N_b, A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_t}$
5. $A \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_t}$

A.13 Kao Chow Authentication v.3

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow B : \{A, B, N_a, K_{ab}, K_t\}_{K_{as}}, \{A, B, N_a, K_{ab}, K_t\}_{K_{bs}}$
3. $B \rightarrow A : \{A, B, N_a, K_{ab}, K_t\}_{K_{as}}, \{N_a, K_{ab}\}_{K_t}, N_b, \{A, B, T, K_{ab}\}_{K_{bs}}$
4. $A \rightarrow B : \{N_b, K_{ab}\}_{K_t}, \{A, B, T, K_{ab}\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv T$ $B \mid\equiv \#(T)$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_t}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K_t}{\longleftrightarrow} B)$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_a$ $A \mid\equiv S \mid\Rightarrow A \overset{K_t}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \overset{K_t}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv A \overset{K_t}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_t}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B, A \overset{K_t}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B, A \overset{K_t}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $B \rightarrow A : \left\{ N_b, A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_t}$
5. $A \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_t}$

A.14 Kerberos V5

(A simplified version from [BAN89])

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \left\{ T_s, L, K_{ab}, B, \{T_s, L, K_{ab}, A, \}_{K_{bs}} \right\}_{K_{as}}$
3. $A \rightarrow B : \{T_s, L, K_{ab}, A, \}_{K_{bs}}, \{A, T_a\}_{K_{ab}}$
4. $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv \#(T_s, L)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv T_a$ $A \mid\equiv \#(T_a)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv \#(T_s, L)$ $B \mid\equiv \#(T_a)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv (T_s, L)$ $S \mid\equiv \#(T_s, L)$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ T_s, L, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
2. $S \rightarrow B : \left\{ T_s, L, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
3. $A \rightarrow B : \left\{ T_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
4. $B \rightarrow A : \left\{ A \mid\sim T_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.15 KSL

1. $A \rightarrow B : N_a, A$
2. $B \rightarrow S : N_a, A, N_b, B$
3. $S \rightarrow B : \{N_b, A, K_{ab}\}_{K_{bs}}, \{N_a, B, K_{ab}\}_{K_{as}}$
4. $B \rightarrow A : \{N_a, B, K_{ab}\}_{K_{as}}, \{T_b, A, K_{ab}\}_{K_{bb}}, N_c, \{N_a\}_{K_{ab}}$
5. $A \rightarrow B : \{N_c\}_{K_{ab}}$
6. $A \rightarrow B : M_a, \{T_b, A, K_{ab}\}_{K_{bb}}$
7. $B \rightarrow A : M_b, \{M_a\}_{K_{ab}}$
8. $A \rightarrow B : \{M_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment In message 4, B generates a new ticket $\{T_b, A, K_{ab}\}_{K_{bb}}$. B will accept this ticket in message 6 before the timestamp expires. This is called repeated authentication. However, BAN logic does not provide appropriate logical notations and postulates for reasoning about repeated authentication. The BAN protocol below generated by our program only deals with the key distribution part of the original protocol and one-off authentication.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $B \rightarrow S : \{N_b\}_{K_{bs}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.16 Lowe Modified KSL

1. $A \rightarrow B : N_a, A$
2. $B \rightarrow S : N_a, A, N_b, B$
3. $S \rightarrow B : \{N_b, A, K_{ab}\}_{K_{bs}}, \{N_a, B, K_{ab}\}_{K_{as}}$
4. $B \rightarrow A : \{N_a, B, K_{ab}\}_{K_{as}}, \{T_b, A, K_{ab}\}_{K_{bb}}, N_c, \{B, N_a\}_{K_{ab}}$
5. $A \rightarrow B : \{N_c\}_{K_{ab}}$
6. $A \rightarrow B : M_a, \{T_b, A, K_{ab}\}_{K_{bb}}$
7. $B \rightarrow A : M_b, \{M_a, B\}_{K_{ab}}$
8. $A \rightarrow B : \{A, M_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $B \rightarrow S : \{N_b\}_{K_{bs}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

Comment Again, we have evolved a protocol to meet only the goals of the key distribution part.

A.17 Neumann Stubblebine

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, \{A, N_a, T_b\}_{K_{bs}}, N_b$
3. $S \rightarrow A : \{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bs}}, N_b$
4. $A \rightarrow B : \{A, K_{ab}, T_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$
5. $A \rightarrow B : M_a, \{A, K_{ab}, T_b\}_{K_{bs}}$
6. $B \rightarrow A : M_b, \{M_a\}_{K_{ab}}$
7. $A \rightarrow B : \{M_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv T_b$ $B \mid\equiv \#(T_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment This protocol provides repeated authentication as well as KSL protocol. We generate only the key distribution part of the original protocol and one-off authentication.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $B \rightarrow S : \{T_b\}_{K_{bs}}$
4. $S \rightarrow B : \left\{ B \mid\sim T_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.18 Hwang Modified Neumann Stubblebine

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, \{A, N_a, T_b, N_b\}_{K_{bs}}$
3. $S \rightarrow A : \{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bs}}, N_b$
4. $A \rightarrow B : \{A, K_{ab}, T_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$
5. $A \rightarrow B : M_a, \{A, K_{ab}, T_b\}_{K_{bs}}$
6. $B \rightarrow A : M_b, \{M_a\}_{K_{ab}}$
7. $A \rightarrow B : \{M_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv T_b$ $B \mid\equiv \#(T_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $B \rightarrow S : \{T_b\}_{K_{bs}}$
4. $S \rightarrow B : \left\{ B \mid\sim T_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

Comment As before, we model only the distribution part of the protocol.

A.19 Needham-Schroeder Public Key

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \{K_b, B\}_{K_s^{-1}}$
3. $A \rightarrow B : \{N_a, A\}_{K_b}$
4. $B \rightarrow S : B, A$
5. $S \rightarrow B : \{K_a, A\}_{K_s^{-1}}$
6. $B \rightarrow A : \{N_a, N_b\}_{K_a}$
7. $A \rightarrow B : \{N_b\}_{K_b}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_s}{\longmapsto} S$ $A \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} S$ $B \mid\equiv \overset{K_s}{\longmapsto} S$ $B \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$ $S \mid\equiv \overset{K_a}{\longmapsto} A$ $S \mid\equiv \overset{K_b}{\longmapsto} B$ $A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$ $B \mid\equiv S \mid\Rightarrow \overset{K_a}{\longmapsto} A$ $A \mid\equiv A \overset{N_a}{\rightleftharpoons} B$ $B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$
	Additional	$A \mid\equiv \#(\overset{K_b}{\longmapsto} B)$ $B \mid\equiv \#(\overset{K_a}{\longmapsto} A)$
Desired Goals	Standard	$A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{N_a}{\rightleftharpoons} B$

Comment The two additional assumptions represent a weakness in the protocol.

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ \left\{ \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}} \right\}_{K_a}$
2. $S \rightarrow B : \left\{ \left\{ \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}} \right\}_{K_b}$
3. $B \rightarrow A : \left\{ \left\{ \overset{K_b}{\longmapsto} B, A \overset{N_b}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$
4. $A \rightarrow B : \left\{ \left\{ \overset{K_a}{\longmapsto} A, A \overset{N_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$

The first two outer encryptions are unnecessary (but our technique is conservative).

A.20 Lowe's fixed version of Needham-Schroeder Public Key

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \{K_b, B\}_{K_s^{-1}}$
3. $A \rightarrow B : \{N_a, A\}_{K_b}$
4. $B \rightarrow S : B, A$
5. $S \rightarrow B : \{K_a, A\}_{K_s^{-1}}$
6. $B \rightarrow A : \{N_a, N_b, B\}_{K_a}$
7. $A \rightarrow B : \{N_b\}_{K_b}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_s}{\longmapsto} S$ $A \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} S$ $B \mid\equiv \overset{K_s}{\longmapsto} S$ $B \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$ $S \mid\equiv \overset{K_a}{\longmapsto} A$ $S \mid\equiv \overset{K_b}{\longmapsto} B$ $A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$ $B \mid\equiv S \mid\Rightarrow \overset{K_a}{\longmapsto} A$ $A \mid\equiv A \overset{N_a}{\rightleftharpoons} B$ $B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$
	Additional	$A \mid\equiv \#(\overset{K_b}{\longmapsto} B)$ $B \mid\equiv \#(\overset{K_a}{\longmapsto} A)$
Desired Goals	Standard	$A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$ $A \mid\equiv B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{N_a}{\rightleftharpoons} B$

Comment Again, the two additional assumptions represent a weakness in the protocol. The first two outer encryptions in the BAN protocol below are unnecessary.

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ \left\{ \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}} \right\}_{K_a}$
2. $S \rightarrow B : \left\{ \left\{ \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}} \right\}_{K_b}$
3. $B \rightarrow A : \left\{ \left\{ \overset{K_b}{\longmapsto} B, A \overset{N_b}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$
4. $A \rightarrow B : \left\{ \left\{ \overset{K_a}{\longmapsto} A, A \overset{N_a}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$

A.21 Needham-Schroeder Symmetric Key

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow A : \left\{ N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}} \right\}_{K_{as}}$
3. $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
4. $B \rightarrow A : \{N_b\}_{K_{ab}}$
5. $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment Obviously, the additional assumption used in this protocol is unusual and leads to the very well-known Denning-Sacco freshness attack. From the BAN protocol below, we can see that the technique avails itself of the dubious additional freshness assumption in much the same way as the original protocol.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $A \rightarrow B : \left\{ N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
5. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.22 Amended Needham-Schroeder Symmetric Key

1. $A \rightarrow B : A$
2. $B \rightarrow A : \{A, N_b\}_{K_{bs}}$
3. $A \rightarrow S : A, B, N_a, \{A, N_b\}_{K_{bs}}$
4. $S \rightarrow A : \left\{N_a, B, K_{ab}, \{K_{ab}, N_b, A\}_{K_{bs}}\right\}_{K_{as}}$
5. $A \rightarrow B : \{K_{ab}, N_b, A\}_{K_{bs}}$
6. $B \rightarrow A : \{N_b\}_{K_{ab}}$
7. $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a\}_{K_{as}}$
2. $S \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $B \rightarrow S : \{N_b\}_{K_{bs}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
5. $B \rightarrow A : \{N_b\}_{K_{ab}}$
6. $A \rightarrow B : \left\{ B \mid\sim N_b, N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
7. $B \rightarrow A : \left\{ A \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.23 Otway Rees

1. $A \rightarrow B : N_c, \{N_a, N_c\}_{K_{as}}$
2. $B \rightarrow S : N_c, \{N_a, N_c\}_{K_{as}}, \{N_b, N_c\}_{K_{bs}}$
3. $S \rightarrow B : N_c, \{N_a, K_{ab}\}_{K_{as}}, \{N_b, K_{ab}\}_{K_{bs}}$
4. $B \rightarrow A : N_c, \{N_a, K_{ab}\}_{K_{as}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $A \mid\equiv N_c$ $A \mid\equiv \#(N_c)$ $A \mid\equiv S \mid\Rightarrow B \mid\sim X$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv N_c$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \mid\sim X$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv N_c$ $B \mid\equiv A \mid\sim N_c$

Comment $A \mid\equiv S \mid\Rightarrow B \mid\sim X$ and $B \mid\equiv S \mid\Rightarrow A \mid\sim X$ indicate the trust that A and B have in the server to forward a message from the other client honestly.

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_a, N_c\}_{K_{as}}$
2. $B \rightarrow S : \{N_b, N_c\}_{K_{bs}}$
3. $S \rightarrow A : \left\{ A \mid\sim N_a, B \mid\sim N_c, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
4. $S \rightarrow B : \left\{ B \mid\sim N_b, A \mid\sim N_c, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$

A.24 SPLICE/AS

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow A : S, \{S, A, N_a, K_b\}_{K_s^{-1}}$
3. $A \rightarrow B : A, B, \left\{A, (T, L), \{N_2\}_{K_b}\right\}_{K_a^{-1}}$
4. $B \rightarrow S : B, A, N_b$
5. $S \rightarrow B : S, \{S, B, N_b, K_a\}_{K_s^{-1}}$
6. $B \rightarrow A : B, A, \{B, N_2 + 1\}_{K_a}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_s}{\longmapsto} S$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_s}{\longmapsto} S$ $S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$ $S \mid\equiv \overset{K_a}{\longmapsto} A$ $S \mid\equiv \overset{K_b}{\longmapsto} B$ $A \mid\equiv (T, L)$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$ $B \mid\equiv S \mid\Rightarrow \overset{K_a}{\longmapsto} A$
	Additional	$A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv \#(A \overset{N_2}{\rightleftharpoons} B)$ $B \mid\equiv A \mid\Rightarrow A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv \#(T, L)$ $S \mid\equiv N_a$ $S \mid\equiv N_b$
Desired Goals	Standard	$B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$
	Additional	$A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$

Comment To model the effect of plaintext message (1) and (4), of the original protocol, we additionally assumed $S \mid\equiv N_a$ and $S \mid\equiv N_b$.

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ \left\{ N_a, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}} \right\}_{K_a}$
2. $S \rightarrow B : \left\{ \left\{ N_b, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}} \right\}_{K_b}$
3. $A \rightarrow B : \left\{ \left\{ (T, L), A \overset{N_2}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$
4. $B \rightarrow A : \left\{ \left\{ A \overset{N_2}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$

A.25 Hwang and Chen Modified SPLICE/AS

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow A : S, \{S, A, N_a, B, K_b\}_{K_s^{-1}}$
3. $A \rightarrow B : A, B, \left\{A, (T, L), \{N_2\}_{K_b}\right\}_{K_a^{-1}}$
4. $B \rightarrow S : B, A, N_b$
5. $S \rightarrow B : S, \{S, B, N_b, A, K_a\}_{K_s^{-1}}$
6. $B \rightarrow A : B, A, \{B, N_2 + 1\}_{K_a}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_s}{\longmapsto} S$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_s}{\longmapsto} S$ $S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$ $S \mid\equiv \overset{K_a}{\longmapsto} A$ $S \mid\equiv \overset{K_b}{\longmapsto} B$ $A \mid\equiv (T, L)$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$ $B \mid\equiv S \mid\Rightarrow \overset{K_a}{\longmapsto} A$
	Additional	$A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv \#(A \overset{N_2}{\rightleftharpoons} B)$ $B \mid\equiv A \mid\Rightarrow A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv \#(T, L)$ $S \mid\equiv N_a$ $S \mid\equiv N_b$
Desired Goals	Standard	$B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$
	Additional	$A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$

Comment Our program gives the same BAN protocol as before.

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ \left\{ N_a, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}} \right\}_{K_a}$
2. $S \rightarrow B : \left\{ \left\{ N_b, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}} \right\}_{K_b}$
3. $A \rightarrow B : \left\{ \left\{ (T, L), A \overset{N_2}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$
4. $B \rightarrow A : \left\{ \left\{ A \overset{N_2}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$

A.26 Clark and Jacob modified modified SPLICE/AS

1. $A \rightarrow S : A, B, N_a$
2. $S \rightarrow A : S, \{S, A, N_a, B, K_b\}_{K_s^{-1}}$
3. $A \rightarrow B : A, B, \left\{(T, L), \{A, N_2\}_{K_b}\right\}_{K_a^{-1}}$
4. $B \rightarrow S : B, A, N_b$
5. $S \rightarrow B : S, \{S, B, N_b, A, K_a\}_{K_s^{-1}}$
6. $B \rightarrow A : B, A, \{N_2 + 1\}_{K_a}$

Assumptions	Standard	$A \mid\equiv \overset{K_a^{-1}}{\longmapsto} A$ $A \mid\equiv \overset{K_s}{\longmapsto} S$ $B \mid\equiv \overset{K_b^{-1}}{\longmapsto} B$ $B \mid\equiv \overset{K_s}{\longmapsto} S$ $S \mid\equiv \overset{K_s^{-1}}{\longmapsto} S$ $S \mid\equiv \overset{K_a}{\longmapsto} A$ $S \mid\equiv \overset{K_b}{\longmapsto} B$ $A \mid\equiv (T, L)$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $A \mid\equiv S \mid\Rightarrow \overset{K_b}{\longmapsto} B$ $B \mid\equiv S \mid\Rightarrow \overset{K_a}{\longmapsto} A$
	Additional	$A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv \#(A \overset{N_2}{\rightleftharpoons} B)$ $B \mid\equiv A \mid\Rightarrow A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv \#(T, L)$ $S \mid\equiv N_a$ $S \mid\equiv N_b$
Desired Goals	Standard	$B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $A \mid\equiv B \mid\equiv A \overset{N_2}{\rightleftharpoons} B$ $B \mid\equiv A \mid\equiv A \overset{N_2}{\rightleftharpoons} B$
	Additional	$A \mid\equiv \overset{K_b}{\longmapsto} B$ $B \mid\equiv \overset{K_a}{\longmapsto} A$

Comment For all SPLICE/AS protocols, our program gives the same BAN protocol.

BAN Protocol generated by our program

1. $S \rightarrow A : \left\{ \left\{ N_a, \overset{K_b}{\longmapsto} B \right\}_{K_s^{-1}} \right\}_{K_a}$
2. $S \rightarrow B : \left\{ \left\{ N_b, \overset{K_a}{\longmapsto} A \right\}_{K_s^{-1}} \right\}_{K_b}$
3. $A \rightarrow B : \left\{ \left\{ (T, L), A \overset{N_2}{\rightleftharpoons} B \right\}_{K_a^{-1}} \right\}_{K_b}$
4. $B \rightarrow A : \left\{ \left\{ A \overset{N_2}{\rightleftharpoons} B \right\}_{K_b^{-1}} \right\}_{K_a}$

A.27 Wide Mouthed Frog

1. $A \rightarrow S : A, \{T_a, B, K_{ab}\}_{K_{as}}$
2. $S \rightarrow B : \{T_s, A, K_{ab}\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv T_a$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv \#(T_s)$ $B \mid\equiv A \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow (A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B)$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv T_s$ $S \mid\equiv \#(T_a)$
Desired Goals	Standard	$B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

BAN Protocol generated by our program

1. $A \rightarrow S : \left\{ T_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
2. $S \rightarrow B : \left\{ T_s, A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$

A.28 Lowe Modified Wide Mouthed Frog

1. $A \rightarrow S : A, \{T_a, B, K_{ab}\}_{K_{as}}$
2. $S \rightarrow B : \{T_s, A, K_{ab}\}_{K_{bs}}$
3. $B \rightarrow A : \{N_b\}_{K_{ab}}$
4. $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv T_a$ $A \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv \#(T_s)$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv A \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow (A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B)$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv T_s$ $S \mid\equiv \#(T_a)$
Desired Goals	Standard	$B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$

Comment The two additional desired goals are for mutual entity authentication. However, the second additional goal is not met by the original protocol unless A can recognise the nonce N_b.

BAN Protocol generated by our program

1. $A \rightarrow S : \left\{ T_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
2. $S \rightarrow B : \left\{ T_s, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
3. $B \rightarrow A : \left\{ A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.29 Woo and Lam Mutual Authentication

1. $P \rightarrow Q : P, N_p$
2. $Q \rightarrow P : Q, N_q$
3. $P \rightarrow Q : \{P, Q, N_p, N_q\}_{K_{ps}}$
4. $Q \rightarrow S : \{P, Q, N_p, N_q\}_{K_{ps}}, \{P, Q, N_p, N_q\}_{K_{qs}}$
5. $S \rightarrow Q : \{Q, N_p, N_q, K_{pq}\}_{K_{ps}}, \{P, N_p, N_q, K_{pq}\}_{K_{qs}}$
6. $Q \rightarrow P : \{Q, N_p, N_q, K_{pq}\}_{K_{ps}}, \{N_p, N_q\}_{K_{pq}}$
7. $P \rightarrow Q : \{N_q\}_{K_{pq}}$

Assumptions	Standard	$P \mid\equiv P \overset{K_{ps}}{\longleftrightarrow} S$ $S \mid\equiv P \overset{K_{ps}}{\longleftrightarrow} S$ $Q \mid\equiv Q \overset{K_{qs}}{\longleftrightarrow} S$ $S \mid\equiv Q \overset{K_{qs}}{\longleftrightarrow} S$ $P \mid\equiv N_p$ $P \mid\equiv \#(N_p)$ $Q \mid\equiv N_q$ $Q \mid\equiv \#(N_q)$ $S \mid\equiv P \overset{K_{pq}}{\longleftrightarrow} Q$ $P \mid\equiv S \mid\Rightarrow P \overset{K_{pq}}{\longleftrightarrow} Q$ $Q \mid\equiv S \mid\Rightarrow P \overset{K_{pq}}{\longleftrightarrow} Q$
Desired Goals	Standard	$P \mid\equiv P \overset{K_{pq}}{\longleftrightarrow} Q$ $Q \mid\equiv P \overset{K_{pq}}{\longleftrightarrow} Q$ $P \mid\equiv Q \mid\equiv P \overset{K_{pq}}{\longleftrightarrow} Q$ $Q \mid\equiv P \mid\equiv P \overset{K_{pq}}{\longleftrightarrow} Q$

BAN Protocol generated by our program

1. $P \rightarrow S : \left\{N_p\right\}_{K_{ps}}$
2. $S \rightarrow P : \left\{P \mid\sim N_p, P \overset{K_{pq}}{\longleftrightarrow} Q\right\}_{K_{ps}}$
3. $Q \rightarrow S : \left\{N_q\right\}_{K_{qs}}$
4. $S \rightarrow Q : \left\{Q \mid\sim N_q, P \overset{K_{pq}}{\longleftrightarrow} Q\right\}_{K_{qs}}$
5. $Q \rightarrow P : \left\{N_q\right\}_{K_{pq}}$
6. $P \rightarrow Q : \left\{Q \mid\sim N_q, N_p, P \overset{K_{pq}}{\longleftrightarrow} Q\right\}_{K_{pq}}$
7. $Q \rightarrow P : \left\{P \mid\sim N_p, P \overset{K_{pq}}{\longleftrightarrow} Q\right\}_{K_{pq}}$

A.30 Woo and Lam π

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{N_b\}_{K_{as}}$
4. $B \rightarrow S : \left\{A, \{N_b\}_{K_{as}}\right\}_{K_{bs}}$
5. $S \rightarrow B : \{N_b\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_b$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_b$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv N_b$

BAN Protocol generated by our program

1. $A \rightarrow S : \{N_b\}_{K_{as}}$
2. $S \rightarrow B : \{A \mid\sim N_b\}_{K_{bs}}$

Comment A can check the format of N_b is that of a nonce and so we can legitimately assume $A \mid\equiv N_b$. A cannot assume $\#(N_b)$. This sort of assumption is made in several subsequent protocols.

A.31 Woo and Lam π 1

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{A, B, N_b\}_{K_{as}}$
4. $B \rightarrow S : \left\{A, B, \{A, B, N_b\}_{K_{as}}\right\}_{K_{bs}}$
5. $S \rightarrow B : \{A, B, N_b\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_b$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_b$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv N_b$

Comment This specification is the same as Protocol A.30.
BAN Protocol generated by our program (same as before):

1. $A \rightarrow S : \{N_b\}_{K_{as}}$
2. $S \rightarrow B : \{A \mid\sim N_b\}_{K_{bs}}$

A.32 Woo and Lam π 2

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{A, N_b\}_{K_{as}}$
4. $B \rightarrow S : \left\{A, \{A, N_b\}_{K_{as}}\right\}_{K_{bs}}$
5. $S \rightarrow B : \{A, N_b\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_b$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_b$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv N_b$

Comment This specification is the same as Protocol A.30.
BAN Protocol generated by our program (same as before):

1. $A \rightarrow S : \{N_b\}_{K_{as}}$
2. $S \rightarrow B : \{A \mid\sim N_b\}_{K_{bs}}$

A.33 Woo and Lam π 3

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{N_b\}_{K_{as}}$
4. $B \rightarrow S : \left\{A, \{N_b\}_{K_{as}}\right\}_{K_{bs}}$
5. $S \rightarrow B : \{A, N_b\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_b$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_b$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv N_b$

Comment This specification is the same as Protocol A.30.
BAN Protocol generated by our program (same as before):

1. $A \rightarrow S : \{N_b\}_{K_{as}}$
2. $S \rightarrow B : \{A \mid\sim N_b\}_{K_{bs}}$

A.34 Woo and Lam π f

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{A, B, N_b\}_{K_{as}}$
4. $B \rightarrow S : \left\{A, B, N_b, \{A, B, N_b\}_{K_{as}}\right\}_{K_{bs}}$
5. $S \rightarrow B : \{A, B, N_b\}_{K_{bs}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $A \mid\equiv N_b$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $B \mid\equiv S \mid\Rightarrow A \mid\sim N_b$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
Desired Goals	Standard	$B \mid\equiv A \mid\equiv N_b$

Comment This specification is the same as Protocol A.30.
BAN Protocol generated by our program (same as before):

1. $A \rightarrow S : \{N_b\}_{K_{as}}$
2. $S \rightarrow B : \{A \mid\sim N_b\}_{K_{bs}}$

A.35 Yahalom

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, \{A, N_a, N_b\}_{K_{bs}}$
3. $S \rightarrow A : \{B, K_{ab}, N_a, N_b\}_{K_{as}}, \{A, K_{ab}\}_{K_{bs}}$
4. $A \rightarrow B : \{A, K_{ab}\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_a$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
	Additional	$S \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv S \mid\Rightarrow \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv A \mid\Rightarrow S \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $A \mid\equiv S \mid\Rightarrow B \mid\sim N_a$ $B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv B \mid\equiv N_a$

Comment The original Yahalom Protocol satisfies all the desired goals. However, we can see that some of its non-standard assumptions are not robust enough (e.g. $B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$). These weak assumptions might cause potential flaws. The assumption above have been taken from the original BAN paper. Also we record $B \mid\equiv N_a$ as an initial assumption to record the effect of message (1) in the original protocol.

BAN Protocol generated by our program

1. $B \rightarrow S : \{N_a, N_b\}_{K_{bs}}$
2. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B, \#(A \overset{K_{ab}}{\longleftrightarrow} B) \right\}_{K_{bs}}$
3. $S \rightarrow A : \left\{ B \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
4. $A \rightarrow B : \left\{ A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
5. $B \rightarrow A : \{N_a\}_{K_{ab}}$

A.36 BAN Simplified Version of Yahalom

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, N_b, \{A, N_a\}_{K_{bs}}$
3. $S \rightarrow A : N_b, \{B, K_{ab}, N_a\}_{K_{as}}, \{A, K_{ab}, N_b\}_{K_{bs}}$
4. $A \rightarrow B : \{A, K_{ab}, N_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$
		$S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$
		$B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
		$S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$
		$S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
		$A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$
		$B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$
		$A \mid\equiv N_a$
		$A \mid\equiv \#(N_a)$
		$B \mid\equiv N_a$
		$B \mid\equiv N_b$
		$B \mid\equiv \#(N_b)$
	Additional	$A \mid\equiv S \mid\Rightarrow B \mid\sim N_a$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
		$B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
		$B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv B \mid\equiv N_a$

Comment We use $B \mid\equiv N_a$ as an initial assumption in our program for the same purpose as it is in Yahalom protocol.

BAN Protocol generated by our program

1. $B \rightarrow S : \{N_a, N_b\}_{K_{bs}}$
2. $S \rightarrow A : \left\{ B \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $B \rightarrow A : \{N_b, N_a\}_{K_{ab}}$
5. $A \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

A.37 Lowe Modified Version of Yahalom

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : \{A, N_a, N_b\}_{K_{bs}}$
3. $S \rightarrow A : \{B, K_{ab}, N_a, N_b\}_{K_{as}}$
4. $S \rightarrow B : \{A, K_{ab}\}_{K_{bs}}$
5. $A \rightarrow B : \{A, B, S, N_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_a$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
	Additional	$S \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv S \mid\Rightarrow \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $B \mid\equiv A \mid\Rightarrow S \mid\equiv \#(A \overset{K_{ab}}{\longleftrightarrow} B)$ $A \mid\equiv S \mid\Rightarrow B \mid\sim N_a$ $B \mid\equiv A \overset{N_b}{\rightleftharpoons} B$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv B \mid\equiv N_a$

Comment All the initial assumptions and desired goals in this protocol are same as they are in original Yahalom protocol. Our program also gives a similar solution to Yahalom as we expect.

BAN Protocol generated by our program

1. $B \rightarrow S : \{N_a, N_b\}_{K_{bs}}$
2. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B, \#(A \overset{K_{ab}}{\longleftrightarrow} B) \right\}_{K_{bs}}$
3. $S \rightarrow A : \left\{ B \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
4. $A \rightarrow B : \left\{ A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$
5. $B \rightarrow A : \{N_a\}_{K_{ab}}$

A.38 Paulson's Strengthened Version of Yahalom

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, N_b, \{A, N_a\}_{K_{bs}}$
3. $S \rightarrow A : N_b, \{B, K_{ab}, N_a\}_{K_{as}}, \{A, B, K_{ab}, N_b\}_{K_{bs}}$
4. $A \rightarrow B : \{A, B, K_{ab}, N_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

Assumptions	Standard	$A \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{as}}{\longleftrightarrow} S$ $B \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv B \overset{K_{bs}}{\longleftrightarrow} S$ $S \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv S \mid\Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$ $A \mid\equiv N_a$ $A \mid\equiv \#(N_a)$ $B \mid\equiv N_a$ $B \mid\equiv N_b$ $B \mid\equiv \#(N_b)$
	Additional	$A \mid\equiv S \mid\Rightarrow B \mid\sim N_a$
Desired Goals	Standard	$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$ $B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B$
	Additional	$A \mid\equiv B \mid\equiv N_a$

Comment All the initial assumptions and desired goals in this protocol are same as they are in BAN modified Yahalom protocol.

BAN Protocol generated by our program

1. $B \rightarrow S : \{N_a, N_b\}_{K_{bs}}$
2. $S \rightarrow A : \left\{ B \mid\sim N_a, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{as}}$
3. $S \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{bs}}$
4. $B \rightarrow A : \{N_b, N_a\}_{K_{ab}}$
5. $A \rightarrow B : \left\{ B \mid\sim N_b, A \overset{K_{ab}}{\longleftrightarrow} B \right\}_{K_{ab}}$

B Description of Simulated Annealing

In 1983 Kirkpatrick et al. [KGV83] proposed *simulated annealing*, a new search technique inspired by the cooling processes of molten metals. It merges hill-climbing with the probabilistic acceptance of non-improving moves to find a good state $S \in State$. The basic algorithm is shown in Figure B.1.

The search starts at some initial state $S_0 \in State$. There is a control parameter $T \in \mathbb{R}^+$ known as the *temperature*. This starts high at T_0 and is gradually lowered, typically by *geometric cooling* (that is, by multiplying by a *cooling factor*, $\alpha \in (0,1)$ at each iteration).

At each temperature, a number *MIL* (Moves in Inner Loop) of moves to new states are attempted. A candidate state Y is randomly selected from the neighbourhood $N(S)$ of the current state. The new state Y is accepted if it is better or only slightly worse than S, as measured by a function $f \in State \to \mathbb{N}$. By 'slightly worse' is meant 'no worse than $T \ln U$ lower'. Here U is a random variable $\in (0,1)$, and so $T \ln U \in (-\infty, 0)$; the smaller T is, the more likely that this term is closer to 0 and eventually improving only improving moves are accepted (that is, the technique reduces to hill climbing).

The algorithm terminates when some stopping criterion is met. Common stopping criteria, and the ones used for the work in this paper, are to stop the search after a fixed number $MaxIL$ of inner loops have been executed, or else when some maximum number MUL of consecutive unproductive inner loops have

$S := S_0$
$T := T_0$
repeat until stopping criterion is met
    repeat *MIL* **times**
        Pick $Y \in N(S)$ with uniform probability
        Pick $U \in (0,1)$ with uniform probability
        if $f(Y) > f(S) + T \ln U$ **then** $S := Y$
    $T := \alpha \times T$

Figure B.1: Basic Simulated Annealing for Maximisation Problems

been executed (that is, without a single move having been accepted).

Generally the best state achieved so far is also recorded (since the search may actually move out of it and subsequently be unable to find a state of similar quality).

There are many improvements to efficiency that can be made, including not generating $U \in (0,1)$ unless $f(Y) < f(S)$.